



Java – Regular Expressions

Presented by



Regular Expressions

- **A regular expression is**
 - **- a special sequence of characters**
 - **- uses a specialized syntax for a matching pattern.**
- **Ex: [abc]- a or b or c**
 - [a-z] – any one lower case alphabet
 - [A-Z] - [a-z] – any one upper case alphabet
 - [7]– the digit 7
 - [98] – 9 or 8
 - [0-9] – any one digit between 0 and 9
 - [abcABC234] – any of these characters
 - [a-zA-Z0-9+@-\$\$] - – any of these characters
- **- to help match or find other strings**



RegEx Pattern and Matcher classes

- Regex classes are present in ***java.util.regex*** package,

- **Pattern class :**

- It is the *compiled version of a regular expression*.
- It is used to define a pattern for the regex engine.

Ex: Pattern p = Pattern.compile(".s");//. represents single character

- **Matcher class:**

- Has methods to perform match operations on a character sequence by interpreting a Pattern.

Ex: Pattern p = Pattern.compile(".s");//. represents single character

Matcher m = p.matcher("as");

boolean b = m.matches();



RegEx How?

- **How Regular Expressions?**
- **^ - Start of the expression**
- **[] - Matching characters of the expression**
- **{min,max} - Number of characters to match**
- **Ex - {3,10} - minimum 3 characters and maximum 10 characters**
- **\$ - End of the expression**
- **Ex: `^[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]{8,15}$`**



RegEx Character Classes

Regex Character classes

No.	Character Class	Description	
1	[abc]		a, b, or c (simple class)
2	[^abc]	Any character except a, b, or c (negation)	
3	[a-zA-Z]	a through z or A through Z, inclusive (range)	
4	[a-d[m-p]]	a through d, or m through p: [a-dm-p] (union)	
5	[a-z&&[def]]	d, e, or f (intersection)	
6	[a-z&&[^bc]]	a through z, except for b and c: [ad-z] (subtraction)	
7	[a-z&&[^m-p]]	a through z, and not m through p: [a-lq-z](subtraction)	



Regex Quantifiers

Regex Quantifiers

The quantifiers specify the number of occurrences of a character.

Regex	Description	
X?	X occurs once or not at all	
X+	X occurs once or more times	
X*	X occurs zero or more times	
X{n}	X occurs n times only	
X{n,}	X occurs n or more times	
X{y,z}		X occurs at least y times but less than z times



RegEx Meta Characters

Regex Metacharacters

The regular expression metacharacters work as shortcuts.

Regex	Description
.	Any character (may or may not match terminator)
\d	Any digits, short of [0-9]
\D	Any non-digit, short for [^0-9]
\s	Any whitespace character, short for [\t\n\x0B\f\r]
\S	Any non-whitespace character, short for [^\s]
\w	Any word character, short for [a-zA-Z_0-9]
\W	Any non-word character, short for [^\w]
\b	A word boundary
\B	A non word boundary



RegEx Meta Characters

Another way of expressions by imposing conditions :

`?=.*` means apply the condition for group of characters

A group of characters must be given like :

`(?=.*\d)` - must contain one digit from 0-9

`(?=.*[a-z])` - must contains one lowercase letter

Rules for password validation :

- It contains at least 8 characters and at most 20 characters.
- It contains at least one digit.
- It contains at least one upper case alphabet.
- It contains at least one lower case alphabet.
- It contains at least one special character which includes `!@#$%^&*()-+=^`.
- It doesn't contain any white space.

String `regex = “^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&*()-+=^])(?=\S+$).{8, 20}$”`



